We claim:

1. An intelligent buffering process to reduce lag in a conference video stream after a data forwarding delay, the stream including a plurality of packets defining either a key frame or a delta frame, each of the key frames representing an entire screen image, each of the delta frames representing image changes relative to a previous frame, the process comprising the steps of:

receiving a newly arrived packet;

determining whether the newly arrived packet is a key frame; and if the packet is a key frame:

detecting whether a buffer has reached a predetermined limit and deleting packets in the buffer if the predetermined limit has been reached; and

adding the newly arrived packet to the buffer.

2. The intelligent buffering process of claim 1, further comprising the step of discarding the newly arrived packet if the packet is a delta frame and if the buffer contents have reached the predetermined limit.

3. The intelligent buffering process of claim 2, further comprising the step of determining whether the previous packet was discarded, and if so, discarding each newly arrived packet which contains delta frame if until another packet containing key frame arrives.

4. The intelligent buffering process of claim 3, further comprising the step of adding the newly arrived packet to the buffer if the packet is a delta frame, if the buffer has not reached the predetermined limit and if the previous frame was not discarded.

5. The intelligent buffering process of claim 1, further comprising the step of forwarding each packet from the buffer in the order received.

6. The intelligent buffering process of claim 5, wherein the forwarding step occurs as soon as downstream data traffic permits.

7. The intelligent buffering process of claim 5, wherein both the receiving step and forwarding step are executed above a network protocol level.

8. The intelligent buffering process of claim 5, wherein after the forwarding step the process includes transmitting the packets over a network.

9. The intelligent buffering process of claim 1, wherein the receiving step includes receiving packets sent from a conference attendee computer over the network.

10. An intelligent buffering process to reduce lag in a conference video stream after a data forwarding delay, the stream including a plurality of packets defining either a key frame or a delta frame, each of the key frames representing an entire screen image, each of the delta frames representing image changes relative to a previous frame, the process comprising the steps of:

receiving a newly arrived packet;

determining whether the newly arrived packet is a delta frame, and if so:

detecting whether the previous packet was discarded,

discarding the newly arrived delta frame if the previous packet was discarded;

detecting whether a buffer have reached a predetermined limit; and

discarding the newly arrived packet if the buffer has reached the predetermined limit.

11. The intelligent buffering process of claim 10, further comprising the step of adding the newly arrived delta frame packet to the buffer if the previous frame was not discarded and if the buffer has not reached the predetermined limit.

12. The intelligent buffering process of claim 10, further comprising:

deleting packets from the buffer if the newly arrived packet is a key frame, and if the buffer has reached the predetermined limit; and

adding the key frame to the buffer.

13. A process for managing a video conference among a plurality of attendee computers, the process comprising:

providing a plurality of buffers at a conference server, each of the buffers corresponding to a respective attendee destination;

receiving a newly arrived packet of a video stream, the packets defining respective key frames and a delta frames, each of the key frames representing an entire screen image, each of the delta frames representing image changes relative to a previous frame; and for each of the buffers:

determining whether the newly arrived packet is a key frame; if the packet is a key frame:

deleting the packets in the buffer if the buffer contents have reached a predetermined limit;

adding the key frame to the buffer; and

forwarding non-deleted packets from the buffer to the corresponding attendee destination.

14. The process of claim 13, independently for each of the buffers the process further comprising the step of discarding the newly arrived frame if the frame is a delta frame and if the buffer contents have reached the predetermined limit.

15. The process of claim 14, independently for each of the buffers the process further comprising the step of determining whether the previous packet was discarded, and if so, discarding each newly arrived delta frame packet if until a new key frame packet arrives.

16. The process of claim 15, independently for each of the buffers the process further comprising the step adding a newly arrived packet to the buffer if the packet is a

delta frame, if the buffer has not reached the predetermined limit and if the previous packet was not discarded.

17. The process of claim 13, wherein after the forwarding step the process includes transmitting the packets over a network to an attendee destination according to a verified-delivery protocol.

18. The process of claim 17, wherein the protocol is TCP/IP.

19. The process of claim 13, wherein the process further comprises providing at least one buffer at at least one of the attendee computers, said receiving, determining, deleting, adding steps being performed by the attendee computer on the packets forwarded from the server.

20. The process of claim 13, wherein the process further comprises providing at least one buffer at at least one of the attendee computers, the attendee computer performing said receiving, determining, deleting, adding and forwarding steps on the packets prior to the receiving of packets by the server.

21. A computer-readable medium having computer-executable instructions for performing a process for managing a buffer, the process comprising:

receiving packets of a video stream data, the packets defining a plurality of key frames and delta frames, each of the key frames representing an entire screen, each of the delta frames representing image changes relative to a previous frame;

determining whether a newly arrived packet contains a key frame; and if the packet is a key frame:

detecting whether the buffer contents have reached a predetermined limit and deleting packets in the buffer if the predetermined limit has been reached; and

adding the key frame to the buffer.

22. The computer-readable medium according to claim 21, the process further comprising discarding each packet not containing a key frame which arrives after the buffer contents have reached said predetermined limit.

23. The computer-readable medium according to claim 21, the process further comprising:

determining whether the previous packet was discarded; and

discarding each packet not containing a key frame which arrives subsequent to a discarded packet until the arrival of a new key frame.